CS 6375: Machine Learning - Project 2
Evaluation of Tree-Based Classifiers and Their Ensembles

Instructor : Tahrima Rahman

Navaneeta Padmakumar/NXP230016

10/20/2025

Hyperparameter Grids

Decision Tree (24 combinations):

```
{
    'criterion': ['gini', 'entropy'],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 10],
    'min_samples_leaf': [1, 4]
}
```

Bagging (16 combinations):

```
{
    'n_estimators': [50, 100],
    'max_samples': [0.7, 1.0],
    'max_features': [0.7, 1.0],
    'bootstrap': [True],
    'bootstrap_features': [False, True]
}
```

Random Forest (144 combinations):

```
{
    'n_estimators': [50, 100, 200],
    'criterion': ['gini', 'entropy'],
    'max_depth': [10, 20, None],
    'min_samples_split': [2, 10],
    'min_samples_leaf': [1, 4],
    'max_features': ['sqrt', 'log2']
}
```

Gradient Boosting (144 combinations):

```
{
    'n_estimators': [50, 100, 200],
    'learning_rate': [0.1, 0.2],
    'max_depth': [3, 5, 7],
    'min_samples_split': [2, 10],
    'min_samples_leaf': [1, 4],
    'subsample': [0.8, 1.0]
}
```

CONSOLIDATED RESULTS TABLES

Table 1: Classification Accuracy on CNF Datasets

Dataset	DecisionTree	Bagging	RandomForest	GradientBoosting
c300_d100	0.6450	0.7900	0.8200	0.8500
c300_d1000	0.6700	0.8705	0.8860	0.9945
c300_d5000	0.7803	0.9412	0.9265	0.9997
c500_d100	0.6800	0.8650	0.9200	0.9300
c500_d1000	0.6765	0.9225	0.9560	0.9985
c500_d5000	0.7899	0.9608	0.9652	0.9999
c1000_d100	0.7150	0.9400	1.0000	0.9700
c1000_d1000	0.7840	0.9730	0.9980	0.9980
c1000_d5000	0.8634	0.9884	0.9982	0.9999
c1500_d100	0.8350	1.0000	1.0000	1.0000
c1500_d1000	0.9150	0.9960	0.9995	1.0000
c1500_d5000	0.9556	0.9988	0.9995	1.0000
c1800_d100	0.9500	0.9850	1.0000	0.9950
c1800_d1000	0.9715	0.9990	1.0000	1.0000
c1800_d5000	0.9873	0.9998	1.0000	1.0000
Average	**0.8146**	**0.9487**	**0.9646**	**0.9824**

Best Performance: GradientBoosting with 98.24% average accuracy

Table 2: F1 Score on CNF Datasets

Dataset	DecisionTree	Bagging	RandomForest	GradientBoosting
c300_d100	0.6603	0.7941	0.8182	0.8585
c300_d1000	0.6680	0.8716	0.8865	0.9945
c300_d5000	0.7884	0.9423	0.9273	0.9997
c500_d100	0.7064	0.8708	0.9208	0.9286
c500_d1000	0.6799	0.9236	0.9565	0.9985
c500_d5000	0.7997	0.9609	0.9655	0.9999
c1000_d100	0.7016	0.9394	1.0000	0.9700
c1000_d1000	0.7956	0.9732	0.9980	0.9980
c1000_d5000	0.8676	0.9884	0.9982	0.9999
c1500_d100	0.8390	1.0000	1.0000	1.0000
c1500_d1000	0.9172	0.9960	0.9995	1.0000
c1500_d5000	0.9560	0.9988	0.9995	1.0000
c1800_d100	0.9510	0.9849	1.0000	0.9950
c1800_d1000	0.9717	0.9990	1.0000	1.0000
c1800_d5000	0.9874	0.9998	1.0000	1.0000
Average	**0.8193**	**0.9495**	**0.9647**	**0.9828**

Best Performance: GradientBoosting with 98.28% average F1 score

Table 3: MNIST Classification Results

Classifier	Test Accuracy	Training Time (minutes)
DecisionTree	0.8841	0.10
Bagging	0.9553	3.69
RandomForest	0.9700	0.31
GradientBoosting	0.9654	33.17

Best Performance: RandomForest with 97.00% accuracy

Table 4: Overall Performance Summary

CNF Datasets (15 datasets average)

Metric	DecisionTree	Bagging	RandomForest	GradientBoosting
Test Accuracy	81.46%	94.87%	96.46%	98.24%
Test F1 Score	81.93%	94.95%	96.47%	98.28%

Best Performance: GradientBoosting excels on CNF datasets

MNIST Dataset

Metric	DecisionTree	Bagging	RandomForest	GradientBoosting
Test Accuracy	88.41%	95.53%	97.00%	96.54%
Training Time	0.10 min	3.69 min	0.31 min	33.17 min

Best Performance: RandomForest achieves highest accuracy on MNIST with efficient training time

Analysis Answers

1. Which classifier achieves the best overall generalization accuracy/F1 score? Explain why.

Gradient Boosting achieves the best overall performance with 98.24% average test accuracy and 98.32% average F1 score on CNF datasets.

Why it performs best:

- **Sequential error correction:** Each tree corrects mistakes from previous trees, leading to better learning of complex Boolean logic patterns
- **Adaptive learning:** The learning rate parameter allows fine-grained control, preventing overfitting while maximizing accuracy
- **Optimal for CNF data:** Boolean satisfiability problems have deterministic patterns that gradient boosting exploits effectively through iterative refinement

2. How does increasing the training data size impact accuracy/F1 score for each classifier?

All classifiers benefit from more training data, but **ensemble methods improve more dramatically:**

Decision Tree:

- c300: 62.5% (d100) → 77.9% (d5000) = +15.4% gain
- c1800: 95.0% (d100) → 98.7% (d5000) = +7.2% gain

Gradient Boosting:

- c300: 85.0% (d100) → 99.97% (d5000) = +14.97% gain
- c500: 93.0% (d100) → 99.99% (d5000) = +6.99% gain

Key observation: Single decision trees plateau quickly, while ensembles (especially Gradient Boosting) continue improving with more data. More training samples allow ensemble methods to learn better feature combinations and reduce variance.

3. How does increasing the number of features (clauses) affect classifier performance?

Counterintuitively, more clauses lead to HIGHER accuracy:

Performance by clause count (d5000 datasets):

- c300: 77.9% (DT), 99.97% (GB)
- c1800: 98.7% (DT), 100% (GB)

Explanation:

- **More constraints = easier classification:** With more clauses, fewer variable assignments satisfy the formula, making the decision boundary clearer
- **Redundant information:** Additional clauses provide multiple ways to identify non-solutions, reinforcing correct predictions
- **Deterministic patterns:** Higher clause counts create more deterministic problems where patterns are easier to learn

The problem becomes **easier**, not harder, with more features because the Boolean formulas become more restrictive.

MNIST Analysis:

Which classifier achieves the highest classification accuracy on MNIST? Provide a brief explanation for its superior performance.

Random Forest achieves the highest accuracy of 97.00% on MNIST, outperforming Decision Tree (88.41%), Bagging (95.53%), and Gradient Boosting (96.54%).

Explanation:

Random Forest excels on MNIST for four key reasons:

1. **Optimal for High-Dimensional Visual Data:** MNIST has 784 pixel features with spatial correlations. Random Forest's feature randomization (selecting random subsets of features for each split) naturally captures different spatial patterns across the ensemble, making it well-suited for image data where different pixel regions contain complementary information.

2. **Effective Feature Selection:** By using 'sqrt' or 'log2' features per split (rather than all 784), Random Forest automatically focuses on the most discriminative pixel regions for each tree. This implicit feature selection reduces noise from irrelevant pixels while capturing essential digit patterns.

3. **Balance Between Bias and Variance:** Random Forest provides better bias-variance tradeoff than Gradient Boosting for MNIST. While Gradient Boosting achieved 96.54%, it required 33.17 minutes of training compared to Random Forest's 0.31 minutes, suggesting Random Forest's parallel tree construction and feature randomization are more efficient for this problem.

4. **Robustness to Pixel Variations:** MNIST digits have natural variations in stroke width, rotation, and position. Random Forest's ensemble of diverse trees, each trained on different feature subsets and bootstrap samples, provides robustness to these variations without overfitting.

Comparison to CNF Performance: Unlike CNF datasets where Gradient Boosting's sequential error correction excelled at learning deterministic Boolean logic, MNIST requires capturing spatial patterns in noisy pixel data. Random Forest's parallel, diversity-focused approach is better suited for this visual recognition task than Gradient Boosting's sequential refinement strategy.